

Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Уральский государственный педагогический университет»
Институт математики, физики, информатики и технологий
Кафедра информационно-коммуникационных технологий в образовании

МОДЕЛЬ ВИРТУАЛЬНОЙ СЕТИ В СРЕДЕ UNETLAB

*Выпускная квалификационная работа
бакалавра по направлению подготовки
09.03.02 – Информационные системы и технологии*

Исполнитель: студент группы БС-41z
Института математики, информатики и ИТ
Волков А.С.

Руководитель: к.п.н., доцент кафедры НИТО
Стариченко Е.Б.

Работа допущена к защите
« ____ » _____ 2017 г.
Зав. кафедрой _____

Екатеринбург – 2017

Реферат

Волков А.С. МОДЕЛЬ ВИРТУАЛЬНОЙ СЕТИ В СРЕДЕ UNETLAB, выпускная квалификационная работа: стр.53, рис., библ., назв.

Ключевые слова: МОДЕЛЬ ВИРТУАЛЬНОЙ СЕТИ, ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ, МОДЕЛИРОВАНИЕ, ПРОЦЕССЫ.

Предмет разработки: мульти-вендорная и многопользовательская платформа UNetLab

Цель работы - на основе современной онлайн-платформы виртуализации сетевого оборудования UNetLab, разработать модель вычислительной сети.

В работе были описаны анализы существующих систем виртуализации сетевого оборудования, спроектированы концепции вычислительных сетей и разработаны их модели. На основе разработанных моделей были проведены исследования их эффективности.

В процессе исследования проводились: анализ существующих систем-аналогов виртуализации сетевого оборудования.

Основные конструктивные, технические и технико-эксплуатационные характеристики: модели сложных вычислительных сетей основываются на использовании оборудования компании Cisco Systems.

Область применения: моделирование сложных вычислительных сетей.

Определения

VLAN (Virtual Local Area Network) – виртуальная локальная сеть.

DHCP (Dynamic Host Configuration Protocol) – протокол динамической настройки узла.

EIGRP (Enhanced Interior Gateway Routing Protocol) – протокол динамической маршрутизации.

NAT (Network Address Translation) — преобразование сетевых адресов.

STP (Spanning Tree Protocol) – протокол остоного дерева.

VPN (Virtual Private Network) – виртуальная частная сеть.

IPsec (сокращение от IP Security) – набор протоколов для обеспечения защиты данных.

GRE (Generic Routing Encapsulation) – общая инкапсуляция маршрутов.

Оглавление

ОПРЕДЕЛЕНИЯ.....	3
ВВЕДЕНИЕ.....	5
ГЛАВА 1. ТЕОРЕТИКО-АНАЛИТИЧЕСКАЯ ЧАСТЬ.....	6
1.1 ОБОСНОВАНИЕ ПОСТАНОВКИ ЗАДАЧИ.....	6
1.2 ВЫБОР МЕТОДА РАЗРАБОТКИ.....	7
1.2.1 Классификация эмуляторов.....	10
1.2.2 Выбор сетевого оборудования.....	10
1.2.3 Выбор эмулятора сетевого оборудования.....	12
ГЛАВА 2. ПРАКТИЧЕСКАЯ (ОРИГИНАЛЬНАЯ) ЧАСТЬ.....	25
2.1 МОДЕЛЬНЫЕ ПРЕДСТАВЛЕНИЯ ОБЪЕКТА РАЗРАБОТКИ.....	25
2.1.1 Процесс моделирования вычислительных сетей в UNetLab.....	27
2.1.2 Анализ архитектуры вычислительных сетей.....	32
2.1.3 Разработка концепций вычислительных сетей.....	34
2.2 ОПИСАНИЕ ИСПОЛЪЗУЕМЫХ ТЕХНОЛОГИЙ.....	36
2.2.1 VLAN.....	36
2.2.2 DHCP.....	37
2.2.3 EIGRP.....	38
2.2.4 NAT.....	38
2.2.5 STP.....	39
2.3 МОДЕЛИРОВАНИЕ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ В UNetLAB.....	40
2.3.1 Описание смоделированных вычислительных сетей.....	41
2.4 РЕЗУЛЬТАТЫ АПРОБАЦИИ, ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ.....	43
ЗАКЛЮЧЕНИЕ.....	45
СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....	46
ПРИЛОЖЕНИЯ.....	50
Приложение 1.....	50
Приложение 2.....	53

Введение

Здесь надо немного порассуждать про необходимость моделирования вообще, как метода познания, и моделирования в виртуальных средах, как современном методе изучения сложных систем.

Актуальность определяется противоречием между необходимостью моделирования современных сложных сетей передачи данных до их внедрения и недостаточным развитием этих технологий.

Объект исследования - среды и системы моделирования работы сетей передачи данных.

Предметом исследования является среда проектирования UNetLab.

Цель исследования состоит в разработке технологии моделирования системы передачи данных на примере построения модели сети предприятия.

В процессе выполнения работы необходимо решить следующие задачи:

Исследовать научную и техническую литературу по моделированию сложных систем.

Произвести анализ существующих систем моделирования работы сетевого оборудования.

Построить модель работы сети передачи данных предприятия в выбранной виртуальной среде.

Глава 1. Теоретико-аналитическая часть

1.1

Обоснование постановки задачи

В современном мире новейших технологий сетевые инженеры нередко сталкиваются с проблемами в своей сфере деятельности. В условиях повсеместной глобализации Интернета, а также интенсивной динамики развития сетевых технологий и технократического общества текущий этап характеризуется стремительным ростом степени мгновенных взаимных коммуникаций и автоматизации производственных процессов. Данный вектор развития способствует появлению услуг и сервисов нового поколения, которые приводят к увеличению объёмов информации, передаваемой по различным каналам связи и, как следствие, повышают требования к реализации и внедрению телекоммуникационных систем (ТКС). В связи с этим на этапе разработки и внедрения, возникает необходимость в получении эксплуатационных и испытательных характеристик вышеупомянутых сетей, с целью их последующего анализа и, как результат, получения общего заключения — соответствуют ли разрабатываемые ТКС ключевым требованиям производительности, безопасности, отказоустойчивости, надёжности и эффективности. Вышеописанная необходимость поднимает очень важную проблему области телекоммуникаций — создание испытательных стендов ТКС на основе парадигмы моделирования. Можно отметить, что самым релевантным вариантом для организации тестовых телекоммуникационных стендов является имитационный метод моделирования. Однако описанный программный инструментарий данного метода на сегодняшний день может считаться неэффективным в силу сложности установки, юстировки и дальнейшей эксплуатации в сравнении с новыми программно-ориентированными комплексами и платформами.

1.2

Выбор метода разработки

Моделирование представляет собой метод научного познания, при использовании которого исследуемый объект заменяется более простым объектом, называемым моделью. Основными разновидностями процесса моделирования можно считать два его вида - математическое и физическое моделирование. При физическом (натурном) моделировании исследуемая система заменяется соответствующей ей другой материальной системой, которая воспроизводит свойства изучаемой системы с сохранением их физической природы. Примером этого вида моделирования может служить пилотная сеть, с помощью которой изучается принципиальная возможность построения сети на основе тех или иных компьютеров, коммуникационных устройств, операционных систем и приложений.

Возможности физического моделирования довольно ограничены. Оно позволяет решать отдельные задачи при задании небольшого количества сочетаний исследуемых параметров системы. Действительно, при натурном моделировании вычислительной сети практически невозможно проверить ее работу для вариантов с использованием различных типов коммуникационных устройств - маршрутизаторов, коммутаторов и т.п. Проверка на практике около десятка разных типов маршрутизаторов связана не только с большими усилиями и временными затратами, но и с немалыми материальными затратами.

Но даже и в тех случаях, когда при оптимизации сети изменяются не типы устройств и операционных систем, а только их параметры, проведение экспериментов в реальном масштабе времени для огромного количества всевозможных сочетаний этих параметров практически невозможно за обозримое время. Даже простое изменение максимального размера пакета в

каком-либо протоколе требует переконфигурирования операционной системы в сотнях компьютеров сети, что требует от администратора сети проведения очень большой работы.

Поэтому, при оптимизации сетей во многих случаях предпочтительным оказывается использование математического моделирования. Математическая модель представляет собой совокупность соотношений (формул, уравнений, неравенств, логических условий), определяющих процесс изменения состояния системы в зависимости от ее параметров, входных сигналов, начальных условий и времени.

Особым классом математических моделей являются имитационные модели. Такие модели представляют собой компьютерную программу, которая шаг за шагом воспроизводит события, происходящие в реальной системе. Применительно к вычислительным сетям их имитационные модели воспроизводят процессы генерации сообщений приложениями, разбиение сообщений на пакеты и кадры определенных протоколов, задержки, связанные с обработкой сообщений, пакетов и кадров внутри операционной системы, процесс получения доступа компьютером к разделяемой сетевой среде, процесс обработки поступающих пакетов маршрутизатором и т.д. При имитационном моделировании сети не требуется приобретать дорогостоящее оборудование - его работы имитируется программами, достаточно точно воспроизводящими все основные особенности и параметры такого оборудования.

Преимуществом имитационных моделей является возможность подмены процесса смены событий в исследуемой системе в реальном масштабе времени на ускоренный процесс смены событий в темпе работы программы. В результате за несколько минут можно воспроизвести работу сети в течение нескольких дней, что дает возможность оценить работу сети в широком диапазоне варьируемых параметров. Результатом работы имитационной модели

являются собранные в ходе наблюдения за протекающими событиями статистические данные о наиболее важных характеристиках сети: временах реакции, коэффициентах использования каналов и узлов, вероятности потерь пакетов и т.п.

Существуют специальные языки имитационного моделирования, которые облегчают процесс создания программной модели по сравнению с использованием универсальных языков программирования. Примерами языков имитационного моделирования могут служить такие языки, как SIMULA, GPSS, SIMDIS.

Существуют также системы имитационного моделирования, которые ориентируются на узкий класс изучаемых систем и позволяют строить модели без программирования.

Анализаторы протоколов незаменимы для исследования реальных сетей, но они не позволяют получать количественные оценки характеристик для еще не существующих сетей, находящихся в стадии проектирования. В этих случаях проектировщики могут использовать средства моделирования, с помощью которых разрабатываются модели, воссоздающие информационные процессы, протекающие в сетях.

Программные эмуляторы сетевого оборудования – это программные продукты, позволяющие соединить в себе функции и параметры реальной вычислительной сети. Они были разработаны для проектирования, моделирования и тестирования работы сети. Большинство эмуляторов достаточно удобны в использовании, так как предоставляют графический интерфейс для управления сетевой инфраструктурой, что бывает намного удобнее чем управление подключениями реальных устройств.

1.2.1 Классификация эмуляторов.

Все эмуляторы сетевого оборудования можно разделить на две основные группы:

1. Аппаратно-реализованные эмуляторы.
2. Программно-реализованные эмуляторы.

К первой группе относят, как правило, узко специализированное оборудование, позволяющее при подключении к нему реального телекоммуникационного оборудования имитировать работу реальной телекоммуникационной сети, либо какой-то ее части (как правило - каналов связи). В аппаратных эмуляторах на аппаратном уровне реализованы процессы, протекающие в реальных сетях - возникновение задержек, потерь пакетов, искажения передаваемых данных и т.п. событий. Основная цель разработки и применения аппаратных эмуляторов - исследование работы реального телекоммуникационного оборудования в различных условиях и при различных характеристиках каналов. Ко второй группе эмуляторов относят специально разработанные программы, позволяющие имитировать работу оборудования и каналов связи, а также работу командных интерфейсов активного сетевого оборудования. Основная цель использования программных эмуляторов - применение в качестве научно-исследовательской деятельности, для постановки научных экспериментов. Также, данные программы часто используются в качестве обучающих систем для подготовки персонала в работе с сетевым оборудованием.

1.1.1

1.2.1 Выбор сетевого оборудования.

Для решения задач создания вычислительных сетей разрабатывается сетевое оборудование различного назначения: коммутатор – сетевое оборудование для объединения компьютеров в одну или несколько локальных сетей; маршрутизатор – устройство, предназначенное для взаимодействия компьютеров, находящихся в разных локальных сетях и предоставления доступа в сеть Интернет; межсетевой экран – устройство, обеспечивающее безопасность в сети и т.д. На сегодняшний день существует множество компаний, производящих сетевое оборудование, и компания Cisco Systems считается безусловным фаворитом на рынке сетевого оборудования (занимает около 70% рынка) и предлагает устройства для создания вычислительных сетей от небольшого офиса до крупных корпораций. Компания Cisco Systems является производителем сетевого оборудования с 1984 года и по сей день является лидером в этой отрасли. Сетевое оборудование компании заметно выделяется на фоне конкурентов и обладает многими достоинствами:

- Надежность – сетевое оборудование, выпускаемое компанией, функционирует на базе операционной системы Cisco IOS и включает в себя огромный спектр настройки и конфигурирования устройства;
- Гибкость – сетевые устройства под управлением Cisco IOS могут одновременно выполнять совершенно различные функции: маршрутизационные, защитные, отладочные и т.д.;
- Интеллектуальность – устройства компании содержат широкий спектр различных технологий и протоколов, как стандартных, так и разработанных собственно компанией Cisco;

- Централизация – для управление устройствами могут использоваться мощные комплексы управления и отладки оборудования, например, такие, как Cisco Security Manager и д.р. Из недостатков, можно лишь выделить стоимость оборудования. Однако, стоит отметить то, что высокая стоимость выпускаемого оборудования компанией Cisco окупается за счет надежности и срока службы данного оборудования. Учитывая широкое распространение сетевого оборудования под управлением Cisco IOS (Internetwork Operating System — Межсетевая Операционная Система), а также высокую стоимость данного оборудования, еще более ясным становится необходимость в применении программных эмуляторов сетевого оборудования для создания моделей вычислительных сетей. Именно на оборудовании данного производителя будут проектироваться будущие модели вычислительных сетей.

1.2.2 Выбор эмулятора сетевого оборудования.

Рассмотрим подробнее наиболее популярные эмуляторы, позволяющие создать виртуальные копии сетевого оборудования производства компании Cisco Systems.

Cisco Packet Tracer

Самым популярным эмулятором сетевого оборудования является Cisco Packet Tracer, это эмулятор, разработанный самой компанией Cisco Systems для обучения начинающих специалистов. Packet Tracer получил большое распространение за счет необходимости его применения для прохождения обучения в рамках программ Cisco Network Academy, сетевой академии, в которой ежегодно проходят обучение десятки тысяч начинающих специалистов. Создание сетевой инфраструктуры и последующая модификация происходят через графический интерфейс, который является интуитивно понятным и наиболее удобным из графических интерфейсов управления, предоставляемых рассматриваемыми программными средствами эмуляции сетевого оборудования. Интерфейс хорошо адаптирован для начинающих специалистов и очень сильно упрощает процесс создания новых сетевых инфраструктур или запуск и настройку необходимых для проведения практических занятий сервисов. Пример интерфейса отображен на рисунке 1.

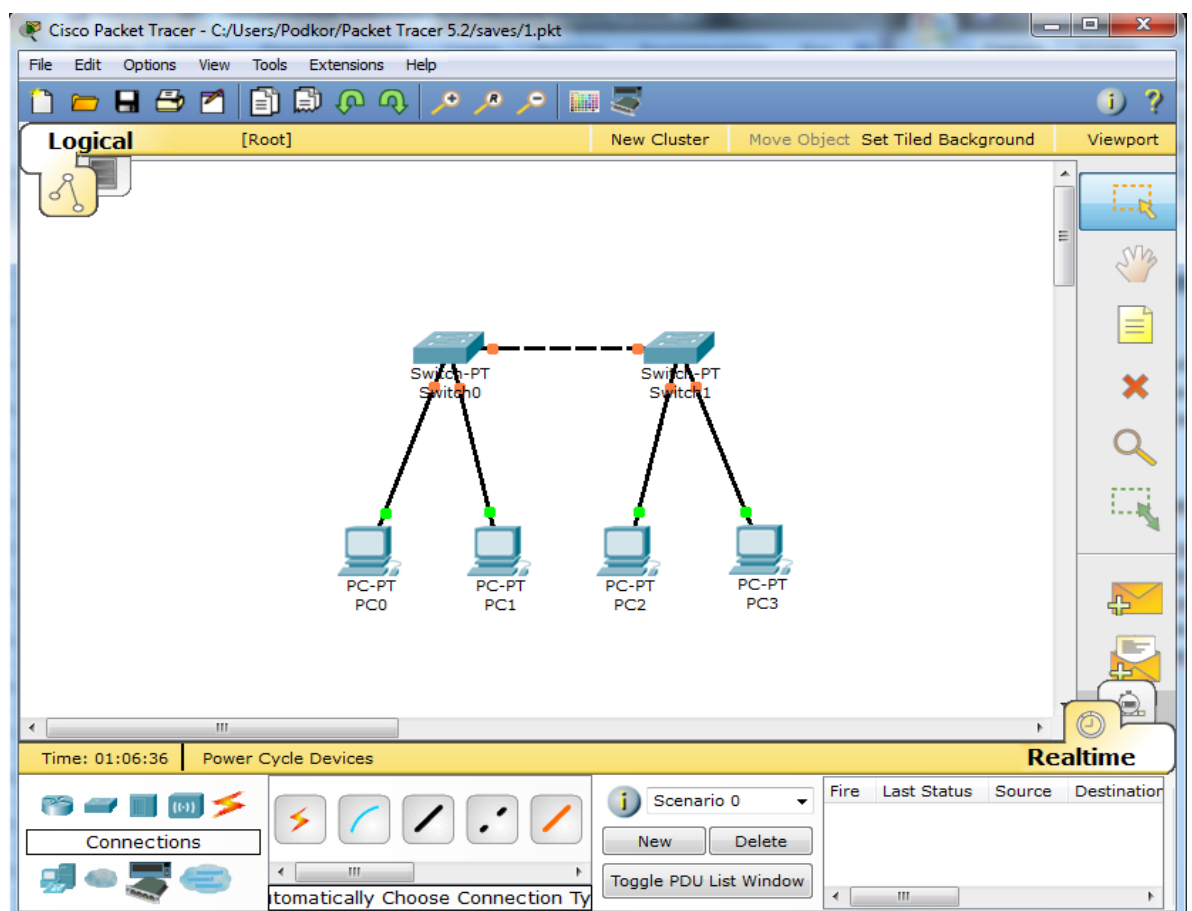


Рисунок 1 - Графический интерфейс эмулятора Cisco Packet Tracer

Основное назначение эмулятора Packet Tracer в создании виртуальных сетей для проведения практических работ для подготовки к сертификационным экзаменам CCNA (Cisco Certified Network Associate) и CCNA Security (Cisco Certified Network Associate Security). Помимо стандартных маршрутизаторов и коммутаторов Packet Tracer поддерживает эмуляцию IP-телефонов, беспроводных точек доступа и серверов с набором стандартных служб. В Packet Tracer встроено множество средств, упрощающих изучение работы сетевой инфраструктуры, таких как снифферы, позволяющие получить подробную информацию о всех блоках данных передаваемых тому или иному устройству, генераторы сетевого трафика, позволяющие искусственно создавать нагрузку, и средства отображения потоков данных, позволяющие проследить маршрут прохождения сети любым пакетом или процесс изменения пакета при прохождении различных устройств. Packet Tracer является удобным средством эмуляции сетевого оборудования не только для обучающегося, но и для преподавателя. В эмулятор встроены средства автоматической проверки выполнения задания. Преподаватель может разработать лабораторную работу для Packet Tracer, которая будет автоматически проверять степень выполнения задания, и вместо проверки вручную правильности работы всех протоколов и корректности введённых команд, достаточно воспользоваться автоматической проверкой, которая определит процент выполнения задания и работоспособность основных сервисов. Cisco Packet Tracer производит эмуляцию как аппаратной, так и программной части сетевого оборудования. Таким образом, Packet Tracer позволяет создавать копии больших сетевых инфраструктур, вот только эмулируемые устройства не поддерживают очень большое количество технологий, используемых в реальных крупных сетях, многие функции, доступные в реальных устройствах попросту отсутствуют. Главное преимущество Cisco Packet Tracer – бесплатность данного продукта. Таким образом, эмулятор Cisco Packet Tracer является оптимальным инструментом для проведения практических занятий при обучении по базовым

курсам компании Cisco и при подготовке к экзаменам уровня специалиста. Но для решения более сложным задач моделирования вычислительных сетей данное ПО не подходит, поскольку является симулятором и не предоставляет всех возможностей реального оборудования, и далее рассматриваться не будет.

Cisco (VIRL) - это программный инструмент, разработанный Cisco для создания и запуска моделирования сети без необходимости в физическом оборудовании.

VIRL представляет собой платформу на базе OpenStack, которая запускает образы программного обеспечения IOSv, IOSvL2, IOS XRv, NX-OSv, CSR1000v и ASAv на встроенном гипервизоре. VIRL обеспечивает масштабируемую, расширяемую сетевую среду проектирования и моделирования, используя интерфейс VM Maestro. В последнее время наблюдается обширная разработка и усовершенствование операций на основе браузера с использованием HTML5. VIRL также имеет широкие возможности для интеграции со сторонними виртуальными машинами, такими как Juniper, Palo Alto Networks, Fortinet, F5 BigIP, Extreme Networks, Arista, Alcatel, Citrix и другие.

VIRL поставляется с полным набором юридических и лицензированных образов Cisco IOS, которые являются такими же, как и на физических маршрутизаторах. Новые выпуски Cisco IOS предоставляются на регулярной основе.

Минимальные требования к оборудованию для VIRL - это компьютер на базе Intel с четырьмя «логическими» ядрами процессора (не физическими процессорами), 8 ГБ оперативной памяти и 70 ГБ свободного места на диске. Большинство последних компьютеров поставляются с процессорами Intel i5 и i7 с четырьмя ядрами (за некоторым исключением). Cisco предлагает большую память, такую как 12 ГБ для 20 узлов, 15 ГБ для 30 узлов или 18 ГБ

для 40 узлов. Для каждого узла Cisco IOS-XRv требуется запуск 3 ГБ памяти. AutoNetkit, который поставляется с VIRL, может автоматически назначать IP-адреса узлам при их запуске, и он даже настроит некоторые основные протоколы маршрутизации. Конфигурация bootstrap дает полностью конвергентную сеть сразу же после ее запуска. Предоставляется возможность сразу перейти к функциям и сосредоточиться на том, что необходимо проверить. Это необходимая функция для сетевых инженеров, позволяющая настроить временную среду для поиска команд и тестирования определенных функций. Если строится топология сети с нуля или создается макет рабочей среды, то рекомендуется ручная IP-адресация. Ниже, на рисунке 2 изображена рабочая область среды проектирования сетей Cisco VIRL.

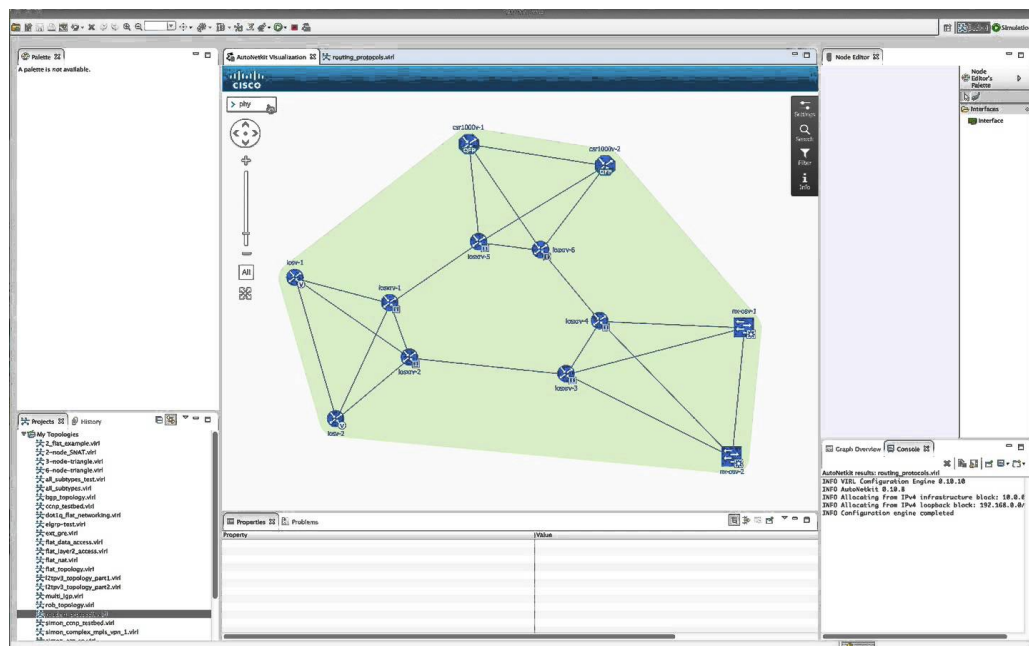


Рисунок 2 – Рабочая область среды моделирования Cisco VIRL

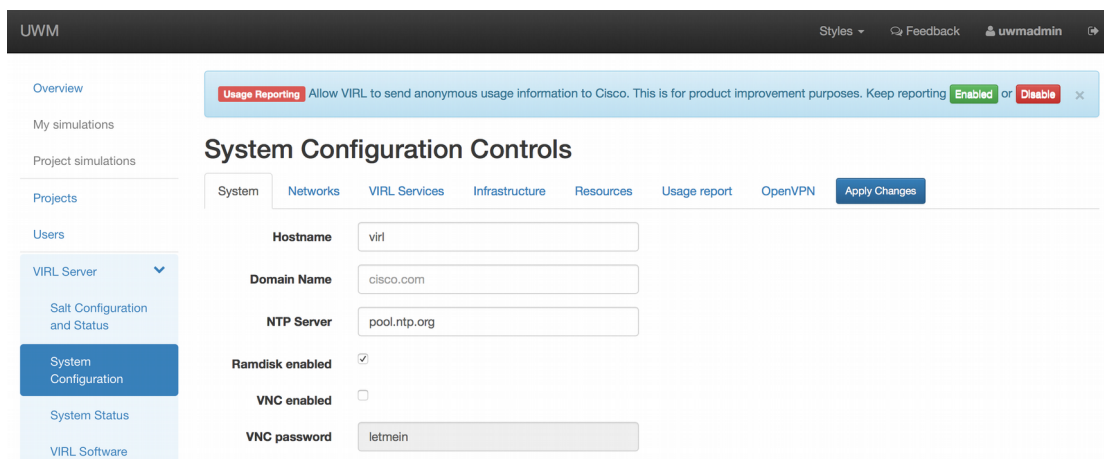


Рисунок 3 – Главное окно среды моделирования Cisco VIRL

GNS3

GNS3 (Graphical Network Simulator 3) – это независимый бесплатный программный эмулятор маршрутизаторов Cisco. GNS3 поддерживается в большинстве операционных систем Linux, Windows и Mac OS X, при этом данный программный эмулятор даёт возможность эмулировать аппаратную часть маршрутизаторов Cisco, для этого он загружает и использует реальный образ операционной системы Cisco IOS.

GNS3 – это графическая оболочка, объединяющая в себе ряд различных программных средств эмуляции. Графический интерфейс среды эмуляции, изображенный на рисунке 4, не адаптирован для начинающих специалистов, он скорее рассчитан на тех, кто уже имеет опыт работы со средствами эмуляции, сетевым оборудованием и знаком с основными принципами функционирования сетевых устройств. Но наличие графических средств управления значительно облегчает процесс создания сетевой инфраструктуры и делает работу с ней более удобной.

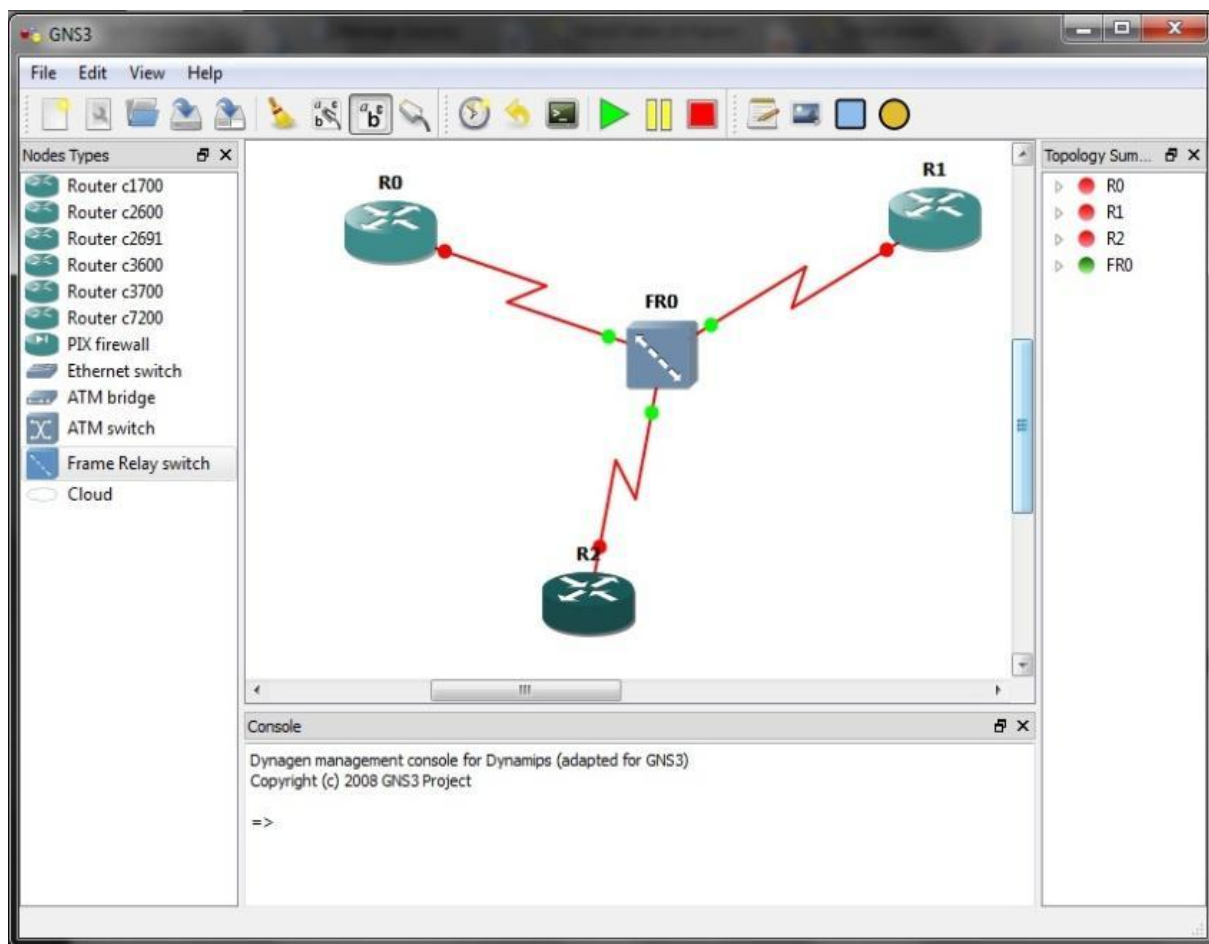


Рисунок 4 - Графический интерфейс эмулятора GNS3

GNS3 включает в себя три отдельных программных эмулятора. Первый из них Dynamips. Многие специалисты, изучающие сетевые технологии, применяют Dynamips исключительно в среде GNS3, так как отпадает необходимость работы с конфигурационными файлами и командной строкой. Вторым является Qemu, который позволяет эмулировать межсетевые экраны Cisco PIX и ASA и системы предотвращения вторжений Cisco IPS, наличие поддержки данных устройств значительно расширяет возможность применения GNS3 в обучении по направлениям, связанным с обеспечением безопасности сетевых инфраструктур. Третьим элементом является система виртуализации VirtualBox, которая позволяет интегрировать в сетевую инфраструктуру из эмулируемых устройств виртуальные сервера или виртуальные персональные компьютеры, которые позволят более точно воссоздать реальную информационную инфраструктуру, а значит изучить больший ряд технологий. GNS3 является очень требовательной к ресурсам системой эмуляции. Так как запускаются одновременно несколько независимых систем эмуляции, а поверх них контролирующая среда, обеспечивающая еще и графический интерфейс, постоянно отображающих изменения в состоянии инфраструктуры, требуются серьезные вычислительные мощности. Хотя GNS3 и дает нам функциональные возможности создать достаточно точную копию реальных информационных инфраструктур с их сетевым, серверным оборудованием и компьютерами конечных пользователей, вычислительной мощности персонального компьютера хватит на эмуляции лишь очень маленькой информационной инфраструктуры. В результате, практические занятия на GNS3 могут проводиться на искусственно созданных сегментах сети, но не на копиях реальных инфраструктур.

UNetLab

Unified Networking Lab (UNetLab, UNL) – сетевой эмулятор, который представляет собой многопользовательскую платформу для моделирования и создания виртуальных сетей, различных лабораторий, поддерживающий внушительный список телекоммуникационного оборудования. Таким образом, концептуальной новизной продукта UNetLab является возможность запуска и использования программы между разными платформами и разными производителями устройств. Пример графического интерфейса отображен на рисунке 5.

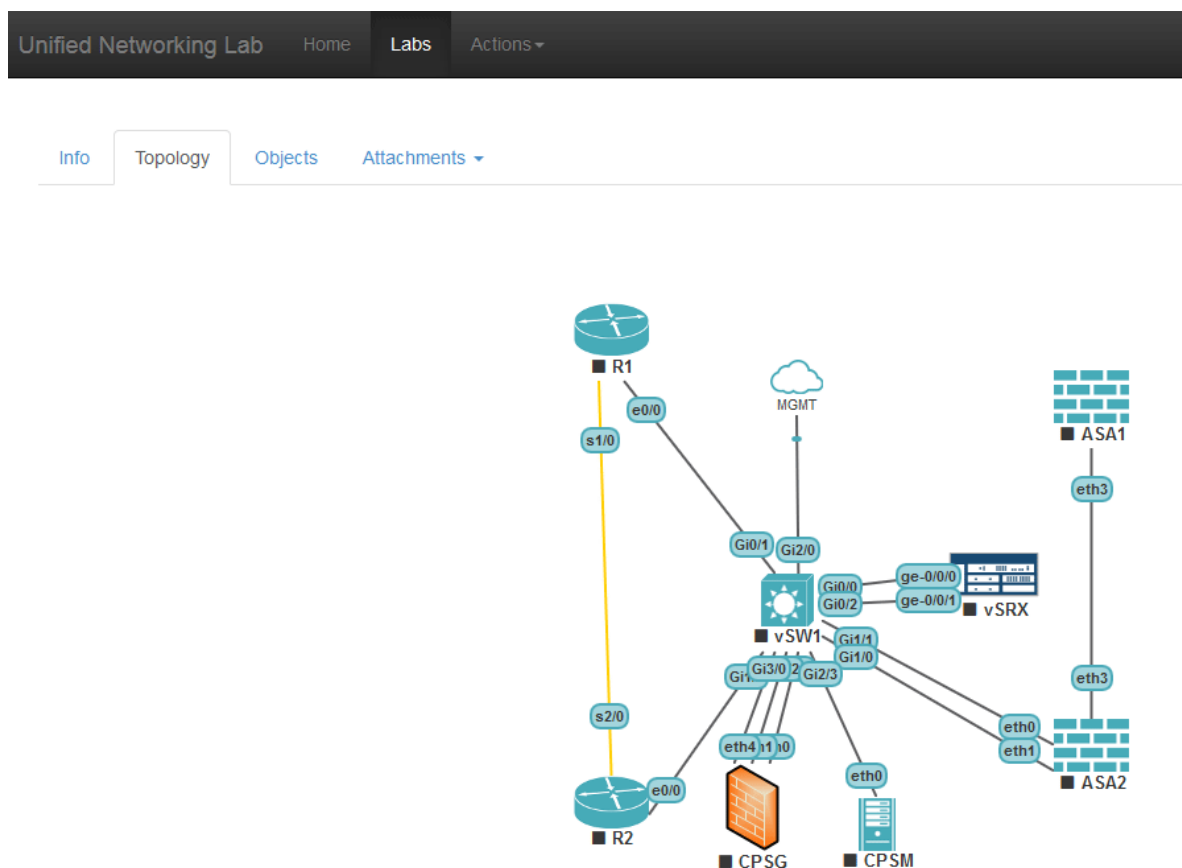


Рисунок 5 - Графический интерфейс эмулятора UNL

В настоящее время эмулятор UNetLab является не только платформой для моделирования виртуальных сетей, но и инструментом для подготовки к различным сертификациям Cisco (для новичков к CCNA/CCNP, так и для профессионалов для подготовки CCIE Routing and Switching, CCIE Security и др.). Кроме того, UNL используется в сетевом инженеринге, в том числе и для системного подхода в выявлении и устранении причин проблемы неполадки сетей (troubleshooting). Проект UNetLab стартовал в марте 2014 года, но за столь короткий срок стал серьезным конкурентом для таких известных эмуляторов как GNS3 и Cisco Packet Tracer, имея в своем багаже ряд огромных преимуществ. При этом разработка продукта осуществляется и по сей день, выявляются ошибки и выходят различные обновления для расширения функционала программы и списка поддерживаемых устройств. Использование данного подхода позволяет UNL отойти от концепции использования автономных виртуальных машин для эмуляции соответствующих сетевых устройств, и создавать цифровые сетевые лаборатории на основе 20 программных эмуляторов IOU/IOL, Dynamips и узлов QEMU, объединяя все необходимые программные модули и сценарии в виде одного файла в рамках одной платформы. Выгодным преимуществом эмулятора UNetLab является то, что он полностью бесплатен, и поэтому может использоваться не только для коммерческих целей, но и для обучения обычными пользователями. Из достоинств так же следует отметить возможность запуска неограниченного количества экземпляров оборудования (роутеров, коммутаторов, устройств безопасности и т.д.), количество ограничено только аппаратными возможностями рабочего места. Поддержка оборудования в UNetLab очень широкая. UNL дает возможность запуска образов из VIRL (vIOS-L2 и vIOS-L3), образов ASA, Cisco IOL-образов, образов Cisco IPS, образов XRv и CSR1000v, образов dynamips из эмулятора GNS, образов Cisco vWLC и vWSA. Кроме перечисленных образов поддерживается внушительный список из оборудования других вендоров: Aruba ClearPass, Alcatel 7750 SR, Arista vEOS,

Brocade Virtual ADX, Citrix Netscaler VPX virtual, Checkpoint Firewall, HP VSR1000, Juniper Olive (porting), Juniper Networks vMX router, Juniper vSRX, S-Terra Firewall, MS Windows и др. Исходя из общего сравнительного анализа программных платформ эмулятора сетевого оборудования, можно выделить UNetLab и GNS3 как наиболее актуальные и эффективные. Следует отметить, что UNetLab в сравнении с GNS3 имеет ряд технических преимуществ, с помощью которых достигается повышение функционала и, как результат, расширение портфеля предоставления услуг в области сетевого проектирования. Сравнительный анализ функциональных характеристик платформ эмулятора сетевого оборудования UNL и GNS3 приведён в таблице 1.

Таблица 1 - Сравнительный анализ функциональных характеристик платформ.

	UNetLab	GNS3
Графический интерфейс	Удобный единый графический интерфейс пользователя на основе технологии WEB автоматически устанавливается вместе с платформой.	Графический интерфейс пользователя в виде специализированного клиента платформы устанавливается пользователем на ПК и отдельно от платформы.
Специализированное ПО	Нет необходимости в отдельных клиентах для использования платформы.	Требуется установки специализированного клиента для последующего использования платформы.
Функциональность	Полноценная поддержка эмуляции канального и сетевого уровней (L2 и L3) без ограничений.	Частичная поддержка эмуляции канального и сетевого уровней (L2 и L3).
Поддержка многопользовательского режима	Многопользовательский функционал, возможность работы нескольких пользователей одновременно.	Строго однопользовательская система.
Ограничения ОЗУ	Нет ограничений ОЗУ под эмуляцию QEMU- устройств.	QEMU поддерживает использование до 2 Гб ОЗУ.

Количество соединений	Отсутствие ограничений по количеству соединений между устройствами в условиях виртуализации QEMU.	Ограничения в 16 соединений между устройствами в рамках виртуализации QEMU.
Масштабируемость	Образы запускаются и работают в рамках одной виртуальной машины или физического сервера	Необходимость в создании отдельных виртуальных машин для запуска образов в GNS3.
Нативная поддержка графических обозначений	Интерфейс пользователя обеспечивает нативную поддержку пользовательских графических обозначений устройств.	Поддержка организации собственных значений устройств частично присутствует.

Исходя из анализа всех вышеперечисленных программных продуктов, явным фаворитом является UNetLab, в силу своего бесплатного распространения, огромного функционала, большого количества поддерживаемых эмулируемых устройств, а также в удобстве создания тестовых стендов сетевого оборудования в проектирования вычислительных сетей, именно UNetLab будет выбран в качестве программного эмулятора сетевого оборудования для разработки моделей вычислительных сетей.

1.3 Формализованное описание технического задания

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

На разработку информационной системы «Модель виртуальной сети в среде UNetLab»

Составлен на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»

1. Общие сведения.

1.1. Название организации-заказчика.

1.2. Модель виртуальной сети в среде UNetLab

- 1.3. Замена изучаемой системы моделью с достаточной точностью описывающей реальную систему и проведение с ней экспериментов с целью получения информации об этой системе.

2. Характеристика области применения продукта.

- 2.1. Процессы и структуры, в которых предполагается использование продукта разработки. Данный продукт может применяться в различных учебных процессах, а также на предприятиях, при построении моделей корпоративной сети.
- 2.2. Разработка данного продукта не требует большого количества персонала. Достаточно одной единицы. Сотрудник должен владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.
- Требуются базовые навыки работы и администрирования Linux
 - Начальный уровень знаний и опыт работы с продуктами VMware
 - Понимание принципов работы FTP, SSH, VNC, HTTP
 - Базовый технический английский

3. Требования к продукту разработки.

3.1. Требования к продукту в целом:

- Возможность оценки пропускной способности сети и ее компонентов;
- сравнение различных вариантов организации вычислительной системы;
- осуществление перспективного прогноза развития вычислительной системы;
- оценивание требуемого количества и производительности серверов в сети;
- сравнение различных вариантов модернизации вычислительной системы;

3.2. Аппаратные требования.

- Процессор **Intel** с полной поддержкой **VT-x/EPT**
- Объем оперативной памяти – 8Гб

3.3. Системное программное обеспечение: ОС Linux, Windows

3.4. Программное обеспечение, используемое для реализации: VMware Workstation Pro, WinSCP, UltraVNC, Putty, Web Browser.

3.5. Входные данные:

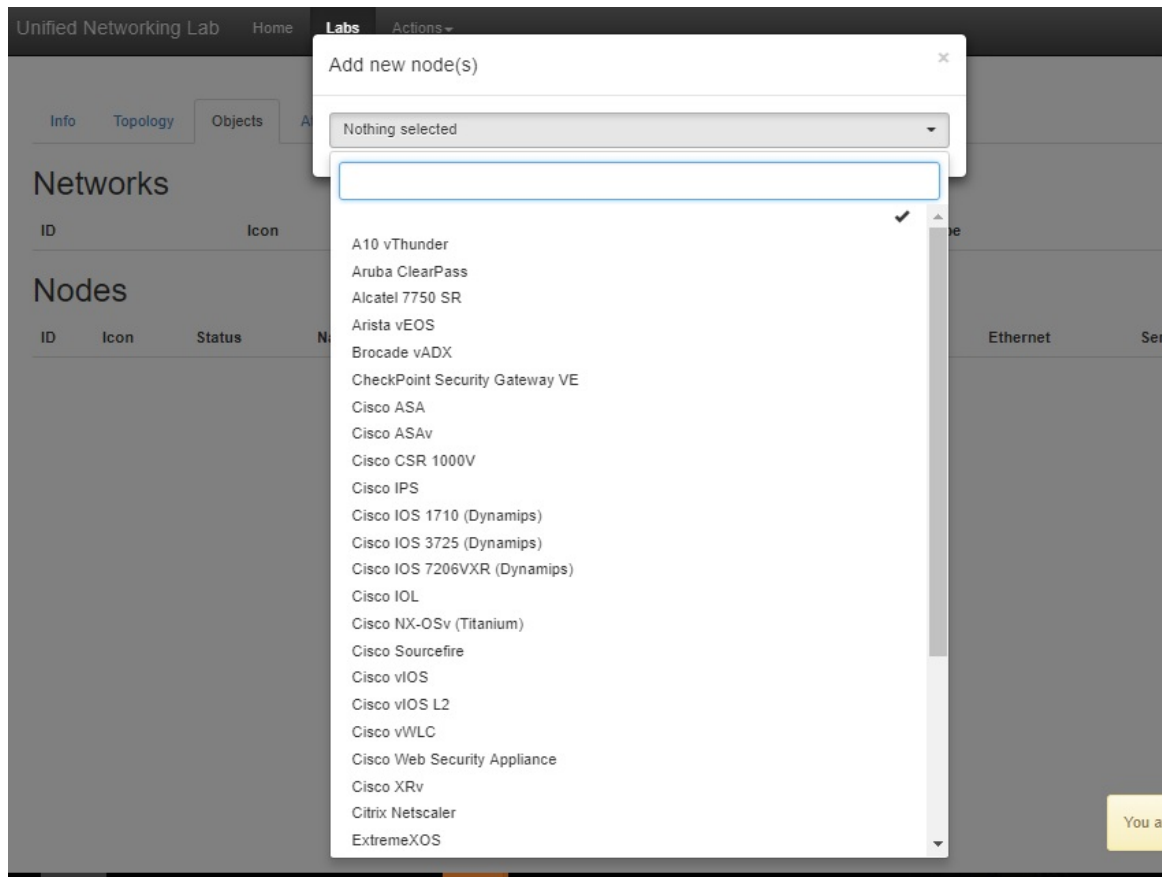


Рисунок 6 – Входные данные UNetLab

Формат входных данных – bin, image, qcow2. Выходных – unl.

3.6. Порядок ввода данных в систему:

Используя ПО WinSCP необходимо перенести образы оборудования в соответствующие папки эмуляторов.

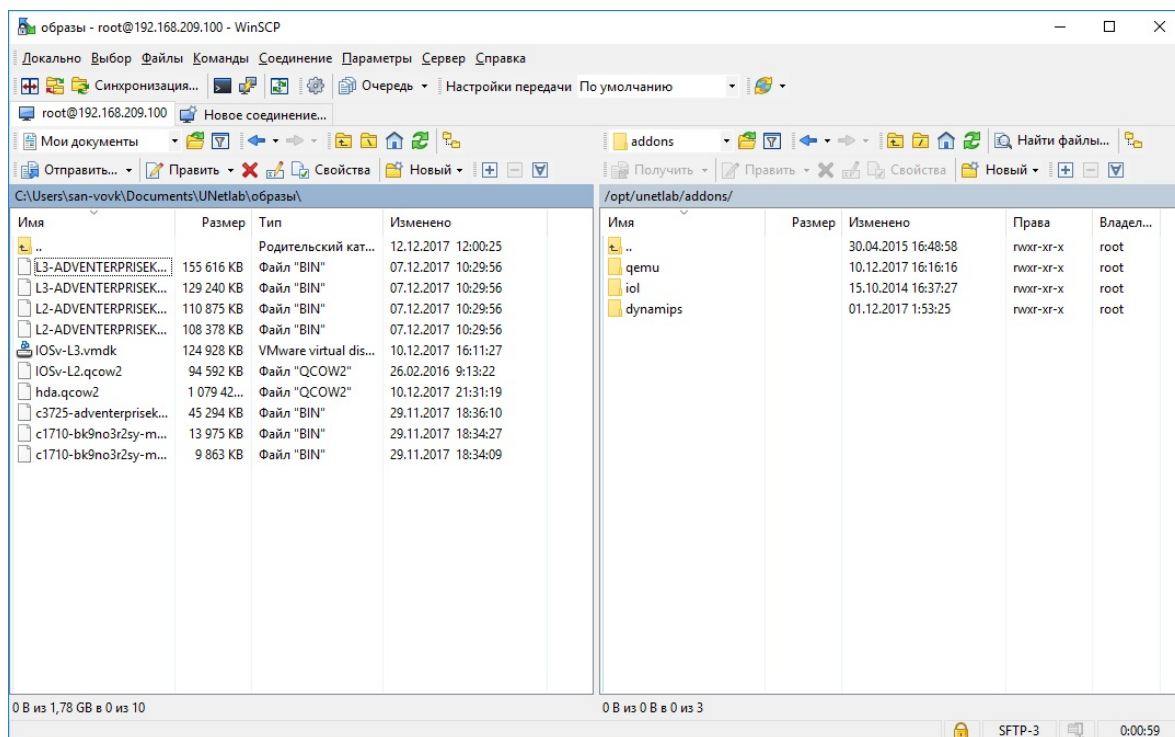


Рисунок 7 – Рабочая область WinSCP

Далее, если это необходимо, конвертировать образы в нужный формат с помощью утилиты Putty.

4. Требования к пользовательскому интерфейсу.

4.1. Пользовательский интерфейс должен отвечать всем необходимым требованиям на сегодняшний день, а именно:

- Доступность
- Минимализм
- Пропускная способность
- Отзывчивость
- Соответствие контексту
- Эффективность

4.2. Размещение информации на экране, дизайн экрана.

Информация на экране должна быть систематизирована, объекты должны располагаться в строгом порядке.

Unified Networking Lab

Home

Labs

Actions

Edit this lab

▶ Start all nodes

■ Stop all nodes

🗑️ Wipe all nodes

📄 Export all CFG

14

Net-14

bridge

15

Net-15

bridge

Nodes











ID	Icon	Status	Name	Type	CPU	Idle PC	NVRAM	RAM	Ethernet	Serial
1		▶	Win-2-2-2-2	qemu	1			256	1	
2		▶	Switch-2	qemu	1			512	16	
3		■	Switch-3	qemu	1			512	16	
4		▶	vIOS	qemu	1			384	4	
5		■	Switch-4	qemu	1			512	16	
6		▶	Win-3-3-3-3	qemu	1			256	1	
7		■	Win-4-4-4-4	qemu	1			256	1	
8		■	Win-5-5-5-5	qemu	1			256	1	
9		▶	Win-6-6-6-6	qemu	1			256	1	
10		■	Win-7-7-7-7	qemu	1			256	1	

Рисунок 8 – Дизайн экрана UNetLab

5. Требования к документированию.

В состав документации модели входит краткое руководство по работе с разработанной моделью, включающее в себя описание состава модели, способы ее реализации и редактирования.

6. Порядок сдачи-приемки продукта.

Вывод по главе 1

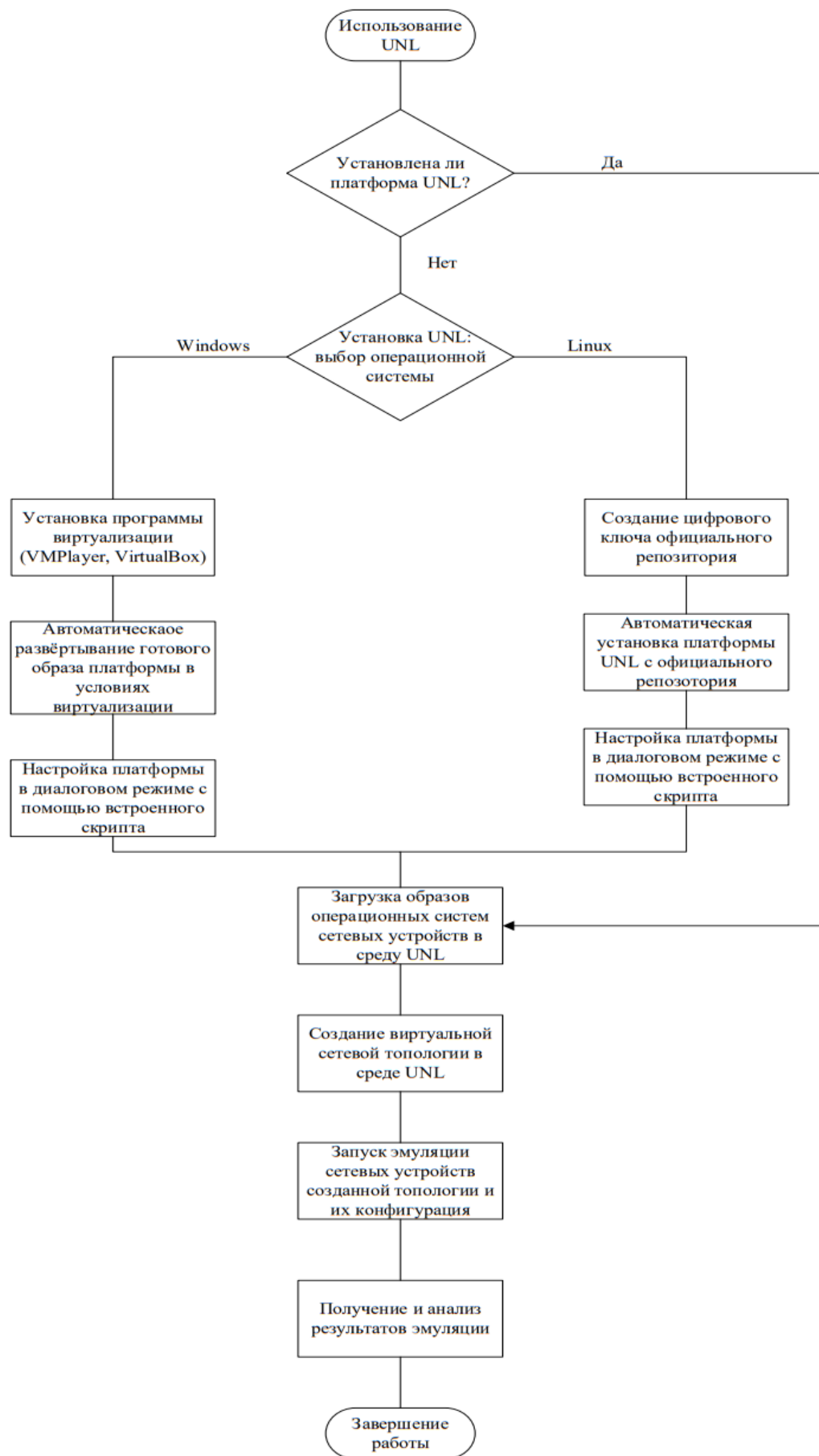
Технологии моделирования существуют и развиваются. Отрабатывать технические решения на моделях безопаснее и дешевле, чем на реальных системах. Для сетей передачи данных существуют как проприетарные среды моделирования, так и свободные. И те и другие имеют положительные черты и недостатки. Наша задача воспользоваться новыми технологиями в этой сфере, применяя открытое программное обеспечение.

Глава 2. Практическая (оригинальная) часть

2.1 Модельные представления объекта разработки

Как и в случае применения других аналогичных платформ, UNetLab предусматривает предварительную установку и юстировку для последующего использования. Общий структурный алгоритм использования UNetLab представлен на рисунке 4.

Руководствуясь структурным алгоритмом, можно выделить следующую хронологию действий при использовании UNetLab: 1. Установка и настройка платформы (опционально): - определение операционной системы, на основе которой разворачивается платформа (Windows/Linux); - установка дополнительного программного обеспечения для виртуализации (актуально при использовании Windows); - установка UNetLab; - полуавтоматическая юстировка платформы в диалоговом режиме с помощью встроенных серверных сценариев. 2. Загрузка образов операционных систем сетевых устройств в UNL. 3. Разработка и создание виртуальной сетевой топологии. 4. Запуск эмуляции сетевых устройств разработанной топологии и их последующая конфигурация. 5. Получение и анализ результатов эмуляции.

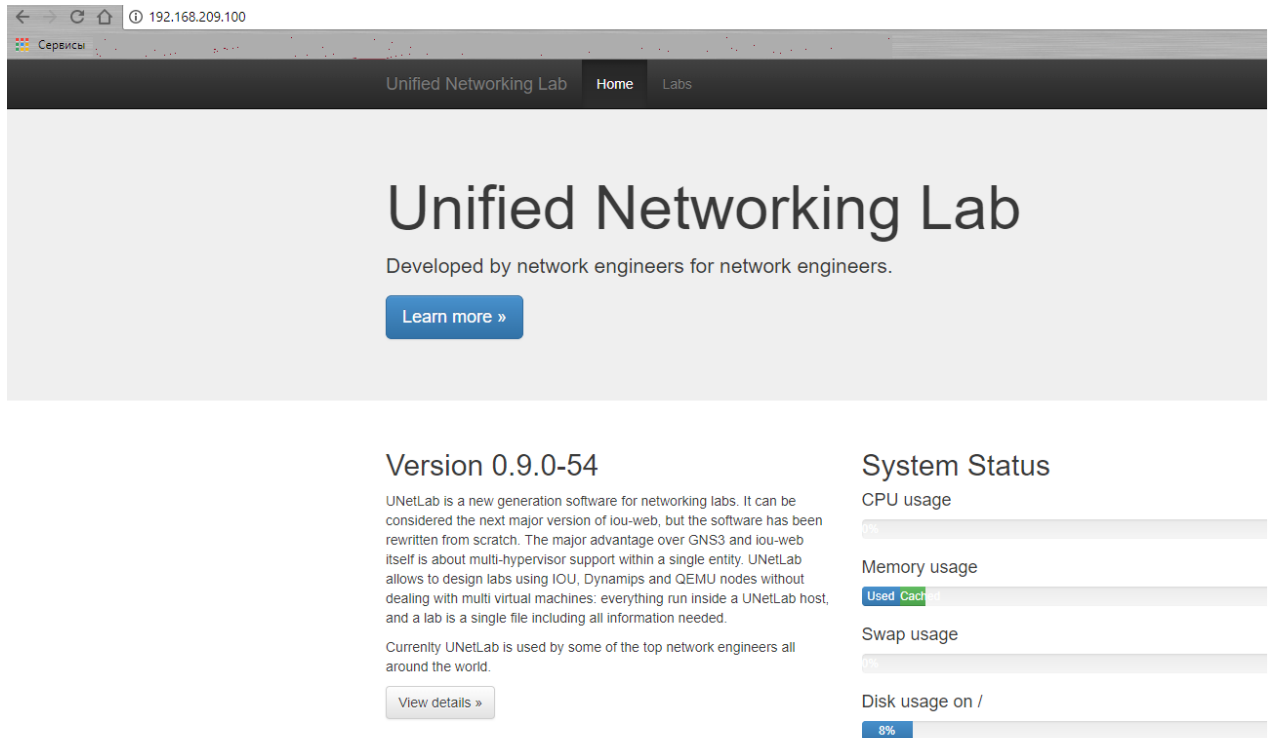


2.1.1 Процесс моделирования вычислительных сетей в UNetLab

Для запуска UNetLab необходимо создать виртуальную машину, на которой и будет развернута наша система. Для создания виртуальной машины использовалось ПО VMware Workstation. Этот программный продукт виртуализации позволяет установить на физический компьютер одну или несколько виртуальных машин. Процесс развертывания UNetLab происходит путем установки исходных файлов на созданной виртуальной машине. По завершению установки, UNetLab готов к работе и становится доступен по IP-адресу, указанному в ходе инсталляции. Процесс моделирования в UNetLab происходит в графическом интерфейсе программы, который становится доступен через веб-браузер. Пользователю необходимо перейти по веб-адресу в браузере, на котором было развернуто программное обеспечение. После чего, он увидит следующее окно аутентификации пользователя, изображенного на рисунке 10.

Рисунок 10 - Аутентификация пользователя в системе UNetLab

После успешной аутентификации, пользователь увидит следующее меню, рисунок 11. На котором изображена главная страница UNetlab, а также текущая статистика используемых ресурсов системой (загрузку ЦП и ОЗУ, количество



используемой памяти и т.д.) и количество запущенных образов.

Рисунок 11 – Главное меню и информация об используемых ресурсах системой UNetLab

Создав новый проект, пользователь попадет на окно рабочего места, рисунок 12. Проектирование будущих вычислительных сетей происходит путем добавления сетевых устройств на рабочую область. Во вкладке “Add an object” пользователю доступен список сетевых устройств различных производителей. Выбрав конкретное устройство, появляется возможность задания характеристик устройства: задание имени, выбор эмулируемого образа устройства, объем оперативной и флэш памяти, количество групп Ethernet портов (в каждой группе по 4 Ethernet порта) и т.д., рисунок 13. Система предоставляет возможность пользователю самому настроить выбранное сетевое устройство. На примере устройств компании Cisco, пользователь выбирает эмулируемый образ сетевого устройства, который эмулирует программную оболочку устройства (Cisco IOS) и путем задания параметров, сам конфигурирует будущее физическое устройство. Таким образом, варьируя параметрами, пользователь может имитировать работу различных, реальных устройств.

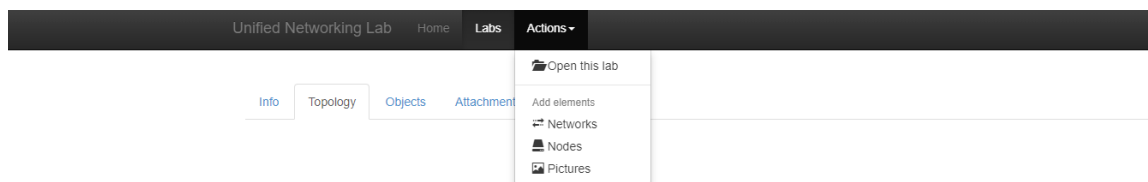


Рисунок 12 - Рабочее место в UNetLab

Add new node(s)

Cisco IOS 1710 (Dynamips)

Number of nodes to add

Image

Name/prefix

Icon

Idle PC

NVRAM

RAM

Startup configuration

Delay (s)

Add **Cancel**

Рисунок 13 - Конфигурирование добавляемого устройства

После того как устройство задано, оно отображается на рабочем поле. Таким образом, пользователь, добавляя на рабочее поле сетевые устройства, конфигурирует будущую архитектуру вычислительной сети. Для связи устройств между собой необходимо выбрать пункт в меню “Interfaces” конкретного устройства и соединить необходимые устройства между собой, таким образом симитировать физическое подключение между ними, рисунок 14.

2.1.2

Анализ архитектуры вычислительных сетей

Одним из главных принципов в архитектуре вычислительных сетей является принцип модульности. Принцип модульности подразумевает то, что всю архитектуру вычислительной сети можно разбить на отдельные модули, что, в свою очередь, позволяет сосредоточиться на функционале каждого модуля по отдельности, при этом, такой подход упрощает ее внедрение и управление. Разбиение большой сети на маленькие модули способствует, в первую очередь, устойчивости сети, так как при возникновении неполадок или сбоев в сети можно локализовать имеющуюся проблему. При этом другие модули сети, которые работают стабильно, не затрагиваются. Еще одним преимуществом модульности сети является возможность упрощенной и безболезненной масштабируемости, которая достигается за счет введения дополнительных модулей при возникающей необходимости расширения вычислительной сети. В архитектуре вычислительных сетей используется иерархическая модель сети, изображенная на рисунке 15, которая впервые была предложена инженерами компании Cisco Systems. Согласно данной модели вычислительная сеть подразделяется на три уровня иерархии, каждый из которых выполняет свою определенную функцию. К уровням иерархической модели относятся: уровень доступа (Access Layer), уровень распределения (Distribution Layer) и уровень ядра или ядро сети (Core Layer).

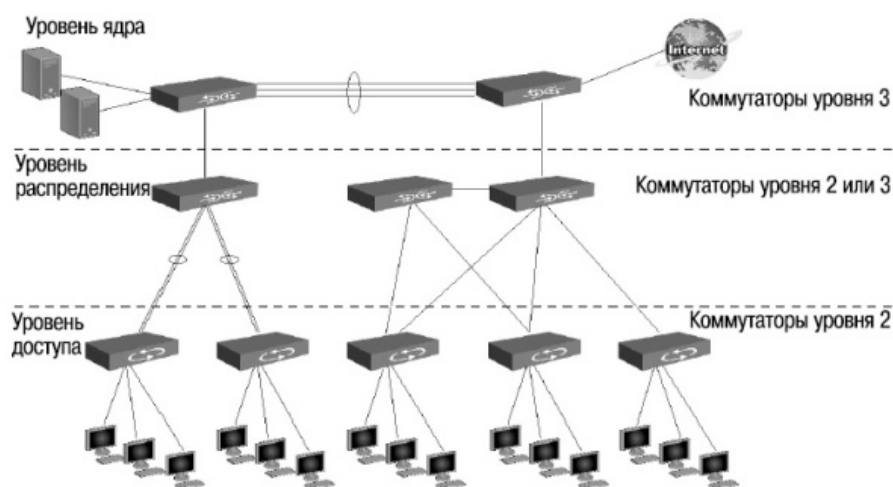


Рисунок 15 - Трехуровневая иерархия вычислительной сети

На уровне доступа (Access Layer) предоставляется доступ к ресурсам сети пользователям или устройствам, таким как сканер, принтер, IP-телефоны и др. Таким образом основная задача этого уровня – это создание точек входа пользователей в объединенную сеть. Уровень доступа зачастую представлен в сети коммутаторами второго уровня (L2) сетевой модели OSI (Open System Interconnection), в редких случаях используются L3-коммутаторы. Следующим уровнем является уровень распределения (Distribution Layer), основной функцией которого является агрегирование уровней доступа и решение задач маршрутизации. На этом уровне используются устройства третьего уровня L3-коммутаторы (маршрутизаторы), осуществляющие маршрутизацию различного трафика между разными сегментами сети. Также на уровне распределения выполняются функции фильтрации и доступа к глобальным сетям. Объединение коммутаторов в одну сеть позволяет уменьшить количество соединений. Уровень ядра (Core Layer) используется, как правило, в больших сетях, объединяющих несколько офисов или зданий. Этот уровень отвечает за быструю и своевременную передачу больших объемов трафика. Кроме того, следует отметить, что уровень ядра объединяет уровни распределения, поэтому отказоустойчивость этого уровня имеет важное значение. Ошибка на 29 уровне ядра будет влиять на всех пользователей сети. Ядро сети представляет собой совокупность мощных коммутаторов и маршрутизаторов. При построении вычислительной сети помимо иерархической структуры сети нужно руководствоваться следующими основными принципами:

- вычислительная сеть должна быть мультисервисной, что предполагает передачу всех типов трафика, используя единые каналы;
- вычислительная сеть должна строиться на базе открытых стандартов и интерфейсов с целью обеспечения возможности наращивания сети и объединения ее с другими сетями;

- принцип минимизации всех расходов, связанных с созданием и эксплуатацией вычислительной сети. Этот принцип подразумевает, что наиболее эффективной с экономической точки зрения будет сеть, использующая коммутацию пакетов, которая позволит эффективно использовать каналы связи. Модульность вычислительной сети, о которой упоминалось выше, предполагает под собой создание отдельных модулей под различные функции. К основным модулям вычислительной сети можно отнести модуль сети Интернет, модуль территориальных сетей и серверный модуль.

2.1.3 Разработка концепций вычислительных сетей

Перед тем, как приступить к разработке моделей вычислительных сетей в UNetLab, необходимо сначала спроектировать концепцию будущих моделей. Спроектированная концепция будет отображать будущую топологию вычислительных сетей. Таким образом будут спроектированы все узлы будущей вычислительной сети. Будет произведено разбиение сегментов на подсети и распределение адресов для них, а также будут спроектированы все соединения между устройствами. Разработка такой концепции позволяет учитывать все нюансы будущих сетей, а также избежать ошибок на этапе моделирования сетей в эмуляторе. Первая концепция, изображенная на рисунке 16, будет отображать простейшую вычислительную сеть, главной целью которой является объединение в единую сеть вычислительные машины предприятия и создание возможности сетевого взаимодействия между ними. Данная сеть будет состоять из нескольких коммутаторов для разбиения группы компьютеров на различные локальные сети, разбиение может происходить по любому принципу (например, по отношению компьютеров к определенному подразделению компании), и маршрутизатора, для обеспечения возможности сетевого взаимодействия между рабочими станциями компаниями, находящимися в разных локальных сетях. Построенная вычислительная сеть данным способом может предполагать различное удаление компьютеров, находящихся в одной локальной сети. Например, на разных этажах одного здания. Таким образом, появляется возможность логического объединения компьютеров на подсети, что значительно упрощает в будущем управление такой сетью и логически структурирует ее.

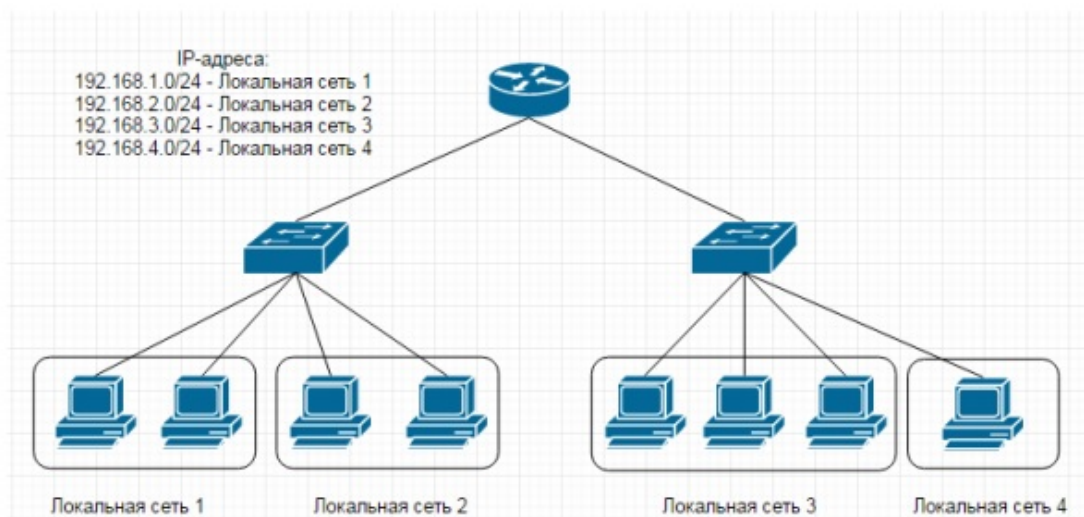


Рисунок 16 - Концепция, отображающая простейшую вычислительную сеть

Следующая концепция, изображенная на рисунке 17, является логическим продолжением предыдущей концепции. На ней продемонстрирован случай, при котором возникает задача логического объединения вычислительных машин в единую локальную сеть, но физически подключенных к разным коммутаторам. В данном случае, компьютеры, логически находящиеся в четвертой локальной сети, но физически подключенные к разным коммутаторам будут иметь возможность сетевого взаимодействия даже без учета маршрутизатора. Однако ситуация с компьютерами логически находящихся в первой локальной сети противоположна, взаимодействие возможно только при наличии маршрутизатора.

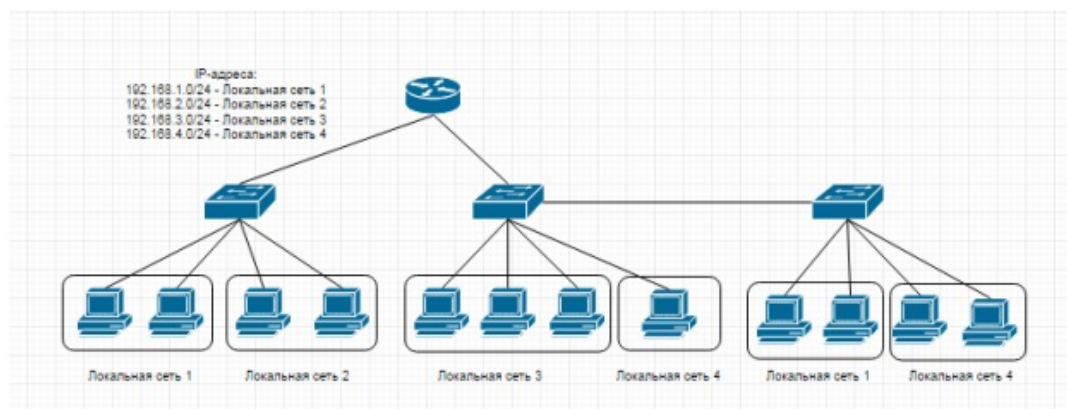


Рисунок 17 - Логическое продолжение предыдущей концепции

2.2

Описание используемых технологий

В разработанных ранее концепциях было сформулировано множество различных задач, от логического разбиения на подсети, формирование IP-адресов, для возможности работы в сети, создание сообщающих тоннелей до возможности выхода в сеть Интернет рабочих станций. Кроме того, для корректного функционирования вычислительной сети, необходимо маршрутизировать весь протекающий трафик внутри сети для возможности сетевого взаимодействия между компьютерами. Для достижения всех этих задач, а также задач по повышению отказоустойчивости и защите информации, протекающей по сообщающим тоннелям, необходимо использовать различные сетевые протоколы, с помощью которых и будет организована вся работа внутри вычислительной сети. Рассмотрим данные технологии поподробнее.

2.2.1

VLAN

Для решения задачи логического структурирования сети будет использоваться технология VLAN (Virtual Local Area Network, виртуальная локальная сеть) — это технология, позволяющая на одном физическом сетевом интерфейсе создавать несколько виртуальных локальных сетей, таким образом, разбивая сеть на логические подсети. Технология позволяет устройствам взаимодействовать между собой напрямую на канальном уровне, хотя физически при этом они могут быть подключены к разным сетевым коммутаторам. И наоборот, устройства, находящиеся в разных VLAN, невидимы друг для друга на канальном уровне, даже если они подключены к одному коммутатору, и связь между этими устройствами возможна только на сетевом и более высоких уровнях, т.е. с использованием маршрутизаторов. Отметим основные достоинства данной технологии:

- Гибкое разделение устройств на группы как правило, одному VLAN соответствует одна подсеть. Компьютеры, находящиеся в разных VLAN, будут изолированы друг от друга;

- Уменьшение широковещательного трафика в сети Каждый VLAN представляет собой отдельный широковещательный домен. Широковещательный трафик не будет транслироваться между разными VLAN;
- Увеличение безопасности и управляемости сети в сети, разбитой на виртуальные подсети, удобно применять политики и правила безопасности для каждого VLAN. Политика будет применена к целой подсети, а не к отдельному устройству;
- Уменьшение количества оборудования и сетевого кабеля для создания новой виртуальной локальной сети не требуется покупка коммутатора и прокладка сетевого кабеля.

2.2.2 DHCP

Для работы компьютера в сети ему необходим IP-адрес. Присваивание IP-адреса компьютеру может быть произведено как статическим методом (ручное задание IP-адреса пользователем), так и динамически (автоматическое присваивание IP-адреса). Поскольку в нашей сети количество рабочих станций может исчисляться десятками, то определенно необходимо использовать именно второй способ. DHCP (Dynamic Host Configuration Protocol — протокол динамической настройки узла) — это сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети. Данный протокол работает по модели «клиент-сервер», где в качестве клиента выступает компьютер, запрашивая у DHCP-сервера конфигурации для работы в сети. Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому серверу DHCP и получает от него нужные параметры. Сетевой администратор может задать диапазон адресов, распределяемых сервером среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок.

2.2.3 EIGRP

Для корректного функционирования вычислительной сети, необходимо маршрутизировать весь протекающий трафик внутри сети для возможности сетевого взаимодействия между компьютерами. Маршрутизация – это процесс определения маршрута в сети. Маршрутизация бывает 2 видов: – статическая маршрутизация; – динамическая маршрутизация. При статической маршрутизации маршруты будут задаваться администратором сети. Данный вид маршрутизации очень удобен для реализации маленькой сети, но непрактичен в большой сети, так как все маршруты задаются при конфигурации маршрутизатора. Сеть, построенная на статической маршрутизации, является неустойчивой, а также плохо масштабируемой. Данный вид маршрутизации очень неэффективен для реализации вычислительной сети для развивающегося предприятия. В сети, настроенной с помощью динамической маршрутизации, таблица маршрутизации редактируется программно, то есть осуществление динамической маршрутизации происходит за счет протоколов маршрутизации.

EIGRP (Enhanced Interior Gateway Routing Protocol) — это протокол динамической маршрутизации, разработанный фирмой Cisco Systems в 1994 году. Принцип работы протокола заключается в трех основных шагах. Сначала маршрутизаторами происходит обнаружение соседних устройств, затем происходит обмен топологической информацией между соседями и в конце маршрутизаторы анализируют полученную информацию и выбирают из нее маршруты с наименьшей метрикой к каждой сети. После того как эти три этапа будут выполнены, в маршрутизаторе будет храниться 3 таблицы: таблица соседних устройств; таблица топологии, полученная от соседних устройств; таблица маршрутизации, с оптимальными маршрутами до всех известных подсетей.

2.2.4 NAT

Для решения задачи доступа в сеть Интернет устройствам компании, через выделенный IP-адрес провайдером, будет использоваться технология NAT. NAT (Network Address Translation — «преобразование сетевых адресов») — это технология в TCP/IP сетях, с помощью которого несколько компьютеров или устройств частной сети (с частными адресами из таких диапазонов, как 192.168.x.x, 172.x.x.x) могут совместно пользоваться одним адресом IPv4, обеспечивающим выход в глобальную сеть. Основная причина растущей популярности NAT связана со все более обостряющимся дефицитом адресов протокола IPv4 — текущего протокола интернета. Отметим основные достоинства данной технологии:

- Экономия публичных IP-адресов через один адрес, можно выпустить больше 65000 серых адресов;
- Препятствует внешним соединениям доходить до конечных компьютеров если извне на устройство с включенной технологией NAT приходит пакет, который не разрешён, он просто отбрасывается;
- Скрывает от посторонних глаз внутреннюю структуру сети при трассировке маршрута извне, ничего далее устройства с включенным NAT доступно не будет;
- Уменьшение количества оборудования и сетевого кабеля для создания новой виртуальной локальной сети не требуется покупка коммутатора и прокладка сетевого кабеля.

2.2.5 STP

Для решения задачи отказоустойчивого доступа к серверу, отображённого в последней концепции, будет использоваться сетевой протокол STP, а именно его улучшенная версия RSTP, версия протокола STP с ускоренной реконфигурацией топологии. STP (Spanning Tree Protocol, протокол остовного дерева) — основная задача STP — предотвратить появление петель на канальном уровне. Работа протокола заключается в блокировке дублирующего маршрута, тем самым предотвращая появление петель. В нашей концепции работа протокола будет заключаться в “резервировании” одного маршрута до сервера. При возникновении неисправности в одном из действующих маршрутов ведущего до сервера, он будет продублирован зарезервированным маршрутом, в другое время, “зарезервированный” маршрут будет заблокирован, во избежание петель в топологии.

2.3 Моделирование вычислительной сети в UNetLab

На основе выбранной современной онлайн-платформы виртуализации сетевого оборудования UNetLab, были разработаны несколько моделей вычислительных сетей. В спроектированных моделях отображен путь становления компании от небольшого офиса до крупной компании, имеющей территориально отдаленные филиалы, нуждающейся в высоких вычислительных ресурсах и их защите.

Для загрузки образов виртуальных машин и сетевого оборудования, а также его дальнейшей настройки использовались встроенные средства UNetLab – QEMU. Были использованы образы L2 и L3 коммутаторов компании Cisco, Виртуальная машина – Windows XP.

Процесс моделирования вычислительных сетей с использованием оборудования компании Cisco Systems происходил путем эмулирования операционной системы Cisco IOS коммутирующего и маршрутизирующего оборудования. Параметры эмулируемых устройств были заданы: для коммутаторов – 128 MByte для оперативной памяти (RAM) и флэш памяти (NVRAM) соответственно; для маршрутизаторов - 128 Mbyte для флэш памяти (NVRAM) и 256 Mbyte для оперативной памяти (RAM).

Таким образом, при физическом воссоздании спроектированных моделей в UNetLab, выбор типа и серии оборудования, может быть обусловлено указанными характеристиками и необходимым количеством портов для подключения оборудования.

В результате, в платформе UNetLab на основе разработанных концепциях вычислительных сетей и использованием выбранный сетевых технологий, были спроектированы модели вычислительных сетей.

2.3.1 Описание смоделированных вычислительных сетей

Модель №1, изображенная на рисунке 18, представляет собой модель, состоящую из двух коммутаторов SWITCH-2 и SWITCH-3 и маршрутизатора vIOS, и отображает простейшую вычислительную сеть, главной целью которой является объединение в единую сеть вычислительные машины и создание возможности сетевого взаимодействия между ними.

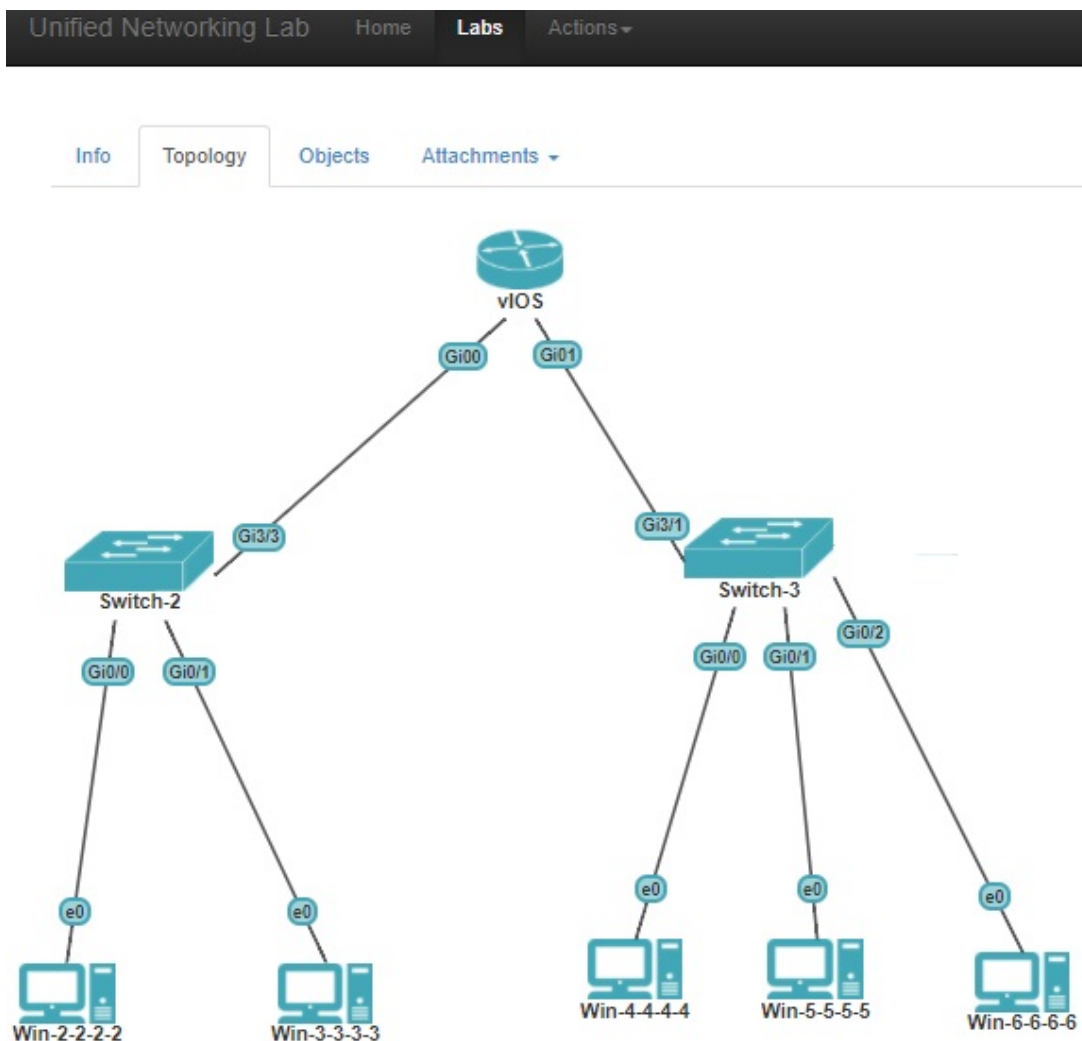


Рисунок 18 - Спроектированная модель вычислительной сети №1

На каждом коммутаторе настроены виртуальные локальные сети (VLAN). На коммутаторе SWITCH-2 настроены локальные сети VLAN2 и VLAN3, а на коммутаторе SWITCH-3 – VLAN4 и VLAN5. Определение отношения рабочих станций к локальным сетям происходит путем распределения на коммутаторе интерфейсов подключения. Данное распределение интерфейсов по локальным сетям на коммутаторе SWITCH-2 продемонстрировано на рисунке 19.

```

GigabitEthernet3/1    unassigned    YES unset    up
GigabitEthernet3/2    unassigned    YES unset    up
GigabitEthernet3/3    unassigned    YES unset    up
Vlan2                 2.2.2.1      YES NVRAM    down
Vlan3                 3.3.3.1      YES NVRAM    down
Switch#
  
```

Рисунок 19 - Распределение интерфейсов подключения по локальным сетям на коммутаторе SWITCH-2

Таким образом, рабочие станции, находящиеся в распоряжении предприятия, были логически распределены в различные локальные сети. Такое деление рабочих станция на локальные сети позволяет упростить в будущем управление такой сетью и логически структурирует ее.

Модель №2, изображенная на рисунке 20, имеет схожую архитектуру вычислительной сети, что и модель №1, поскольку является логическим продолжение предыдущей.

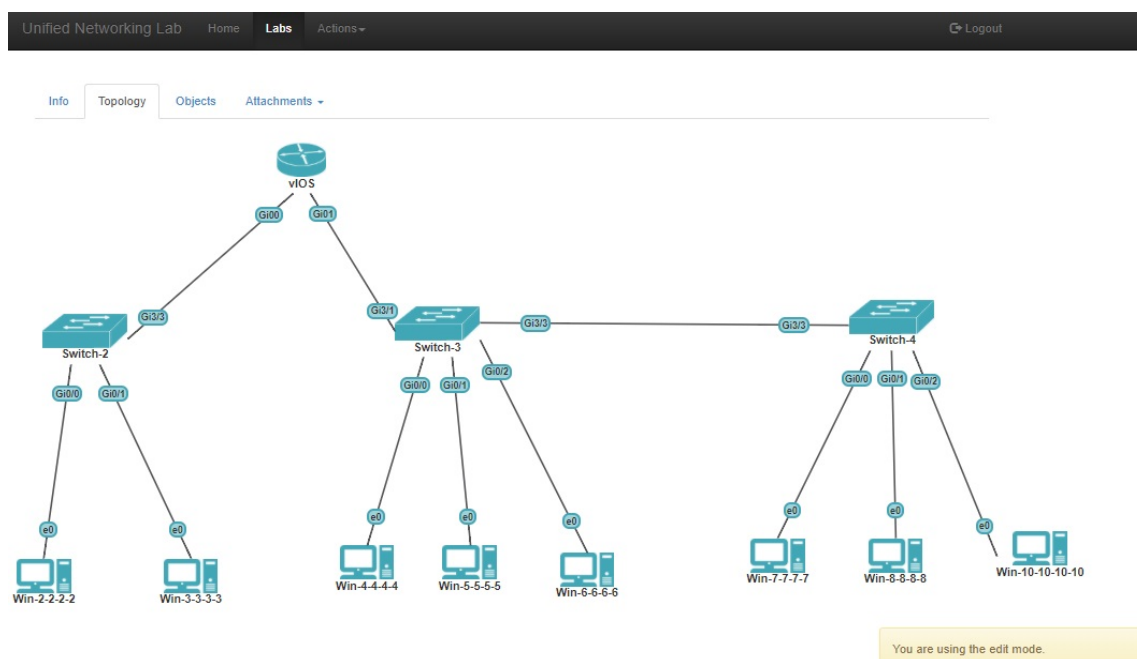


Рисунок 20 - Спроектированная модель вычислительной сети №2

На данной модели продемонстрирован случай, при котором возникает задача логического объединения вычислительных машин в единую локальную сеть, но физически подключенных к разным коммутаторам. В данном случае, компьютеры, логически находящиеся в четвертой локальной сети, но физически подключенные к разным коммутаторам (SWITCH-3 и SWITCH-4) будут иметь возможность сетевого взаимодействия даже без учета маршрутизатора vIOS. Однако ситуация с компьютерами логически находящихся во второй локальной сети противоположна, взаимодействие возможно только при наличии маршрутизатора. Такое логическое объединение стало возможным путем соединения коммутаторов SWITCH-3 и SWITCH-4 магистральным портом (Trunk port). Данный магистральный порт служит для передачи трафика локальных сетей (VLAN) между устройствами.

Для моделей №1 и №2, основной задачей которых является объединение в единую сеть вычислительных машин и создание возможности сетевого взаимодействия между ними, были проведены исследования возможностей этого взаимодействия. Возможность сетевого взаимодействия определялась успешным выполнением сетевой утилиты для проверки целостности и качества сетевого соединения – «ping» между компьютерами одной локальной сети и компьютерами, находящимися в разных локальных сетях.

2.3

Результаты апробации, техническая документация

Инструкции и рекомендации по применению разработанного продукта; результаты апробации. Ссылка на акты внедрения.

При моделировании работы оборудования Cisco в платформе UNetLab, происходит эмуляция только программного обеспечения, оболочки Cisco IOS, а все эмуляция аппаратного обеспечения ложится на оборудовании вычислительной машины, на которой данная платформа установлена. Таким образом, при физическом воссоздании спроектированных моделей в UNetLab, выбор типа и серии оборудования, может быть обусловлено заданными характеристиками при моделировании работы и необходимым количеством портов для подключения оборудования.

Заключение

В ходе выпускной квалификационной работы были решены следующие задачи: исследованы литературные и научные источники, касающиеся моделирования сложных систем; проведён анализ существующих систем моделирования работы сетевого оборудования; на основе современной онлайн-платформы виртуализации сетевого оборудования UNetLab разработана модель функционирования сети передачи данных предприятия.

Сравнительный анализ виртуальных сред показал возможность их использования и эффективность применения для моделирования сетей передачи данных. Выбранная система отвечает требованиям технического задания и, учитывая свободный характер, может быть рекомендована не только для отладки работы сетей передачи данных, но и для использования в учебном процессе.

Таким образом можно сделать вывод, что задачи исследования решены, цель достигнута.

Список информационных источников

1. Razvan, Beuran Intorduction to network emulation. Taylor & Francis Group, 2012. 389 с.
2. А. Марк, Д. Спортак. Компьютерные сети. Книга 1: High-Performance Networking. Энциклопедия пользователя: пер. с англ. К.: ДиаСофт, 1999. 432 с.
3. А. Марк, Д. Спортак. Компьютерные сети. Книга 2: High-Performance Networking. Энциклопедия пользователя: пер. с англ. К.: ДиаСофт, 1999. 432 с.
4. А. Мысник. Локальные сети: архитектура, алгоритмы, проектирование. 3 изд. М.: Вильямс, 2005. 1168 с.
5. А.М. Горячев. Тестирование и применение эмуляторов Cisco для моделирования гетерогенной IP сети. 3 изд. М.: Гагаринские чтения – 2016: XLII Международная молодежная научная конференция: Сборник тезисов докладов Т.ё. Московский авиационный институт (национальный исследовательский университет), 2016. 278 с.
6. В. Шаньгин. Информационная безопасность компьютерных систем и сетей. Инфра-М, 2011. 416 с.
7. В.Г. Олифер, Н.А. Олифе. Компьютерные сети. Принципы, технологии, протоколы. СПб.: Питер, 2016. 992 с.
8. Д. Девис, Д. Барбер, У. Прайс. Вычислительные сети и сетевые протоколы. М.: Мир, 1982. 562 с.
9. Е.Ф. Попов Использование программных средств эмуляции оборудования при модификации сетевой инфраструктуры. Тюмень: Сборник научных трудов по материалам всероссийскую научно-практической конференции студентов, аспирантов и молодых ученых «Новые технологии – нефтегазовому региону, 2012. 258 с.

10. Е.Ф. Попов, А.А. Захаров. Сборник научных трудов по материалам Международной заочной научно-практической конференции «Теоретические и прикладные проблемы науки и образования в 21 веке». Часть 8. Тамбов: ТРОО «Бизнес-Наука- Общество», 2012. 418 с.
11. С.В. Запечников. Основы построения виртуальных частных сетей. 3 изд. М.: М., 2003. 249 с.
12. Т.И. Иванова. Корпоративные сети связи. М.: Пособие, 2001. 297 с.
13. Т. Лэммсл, Ш. Одом, К. Уоллес. CCNP маршрутизация. Лори, 2015. 444 с.
14. Шеннон Р. Ю. Имитационное моделирование систем – искусство и наука: Пер. с англ. – М.: Мир, 1978. 418 с.
15. Э. Таненбаум Компьютерные сети. СПб: Питер, 2002. 248 с.
16. Ю.В. Новиков, С.В. Кондратенко. Локальные сети: архитектура, алгоритмы, проектирование. М.: ЭКОМ, 2001. 312 с.
17. A Technical Overview // Introduction to Cisco IOS Netflow URL: http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/iosnetflow/prod_white_paper0900aecd80406232.html (дата обращения: 12.06.2017).
18. CiscoIOSFlexibleNetFlow URL: <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/fnetflow/configuration/15-mt/fnf-15-mt-book/fnf-fnetflow.html> (дата обращения: 12.06.2017).
19. GNS3 - Графический Сетевой Симулятор // CiscoLab URL: <http://www.ciscolab.ru/labs/40-gns3-graficheskiy-setevoysimulyator.html> (дата обращения: 09.06.2017).
20. Unetlab – отличная замена GNS3 и Cisco Packet Tracer // Litl-admin.ru URL: <https://litl-admin.ru/cisco/unetlab-otlichnaya-zamena-gns3-i-cisco-packet-tracer.html> (дата обращения: 01.06.2017).

21. UNetLab - строим стенд для подготовки CCIE/RS/DC/SP/SEC // Rover's Home URL: <http://rover-seti.blogspot.ru/2015/07/unetlab-ccie-rsspdsec.html> (дата обращения: 17.06.2017).
22. UNetLab. Обзор интерфейса и создание лабораторной работы // Black-Box URL: <http://it-blackbox.blogspot.ru/2016/02/unetlab.html> (дата обращения: 03.06.2017).
23. UNetLab: List of supported images URL: <http://www.unetlab.com/documentation/supported-images/index.html> (дата обращения: 12.06.2017).
24. VIRL - Virtual Internet Routing Lab – Cisco // Cisco Systems URL: <http://virl.cisco.com/> (дата обращения: 09.06.2017).
25. VLAN // Дневник сетевого специалиста URL: <http://day-book.org.ua/seti/chto-takoe-vlan.html> (дата обращения: 09.07.2017).
26. Архитектура корпоративных сетей // NetSkills URL: <http://blog.netskills.ru/p/blog-page.html> (дата обращения: 20.06.2017).
27. Заметки сетевого инженера // BlogSpot URL: <http://gh05ter.blogspot.ru/2016/04/unetlab.html> (дата обращения: 11.06.2017).
28. Курс UNetLab // NetSkills. Online Network School URL: <http://blog.netskills.ru/p/unetlab.html> (дата обращения: 10.06.2017).
29. Курс молодого бойца. Практический курс с использованием Cisco Packet Tracer // NetSkills URL: <http://blog.netskills.ru/p/cisco-packettracer.html> (дата обращения: 22.06.2017).
30. Мировой рынок 2016: перемены в тройке крупнейших производителей // Ethernet-коммутаторы локальных сетей URL: <http://www.tadviser.ru/index.php>. (дата обращения: 15.06.2017).
31. Сертификат CCNA // Cisco Systems URL: http://www.cisco.com/web/RU/learning/le3/le2/le0/le9/learning_certification_type_home.html (дата обращения: 06.06.2017).

32. Сравнение UNetLab с VIRL и GNS3 // CCIENetLab - Сетевая Лаборатория URL: <http://www.ccienetlab.com/labs/9-sravnenie-unetlab-s-virl-i-gns3.html> (дата обращения: 06.06.2017).
33. Эмулятор UNetLab: Запуск, настройка и запуск Cisco L3/L2 IOU // wordpress URL: <https://zotovp.wordpress.com/2016/01/27/-unetlab-cisco.html> (дата обращения: 21.06.2017).

Приложения

Приложение 1.

Настройка первого коммутатора

```
Switch1>en
Switch1#conf t
Switch1(config)#
Switch1(config)#vlan 2
Switch1(config-vlan)#name D
Switch1(config-vlan)#vlan 3
Switch1(config-vlan)#name V
Switch1(config-vlan)#exit
Switch1(config)#
Switch1(config)#interface range fastEthernet 0/2-11
Switch1(config-if-range)#switchport mode access
Switch1(config-if-range)#switchport access vlan 3
Switch1(config-if-range)#switchport voice vlan 4
Switch1(config-if-range)#exit
Switch1(config)#interface fastEthernet 0/1//соединение с маршрутизатором
Switch1(config-if)#switchport mode trunk
Switch1(config-if)#exit
Switch1(config)#
Switch1(config)#exit
Switch1#wr
```

Настройка второго коммутатора:

```
Switch2>en
Switch2#conf t
```

```

Switch2(config)#
Switch2(config)#vlan 2
Switch2(config-vlan)#name D
Switch2(config-vlan)#vlan 3
Switch2(config-vlan)#name V
Switch2(config-vlan)#exit
Switch2(config)#
Switch2(config)#interface range fastEthernet 0/1-10
Switch2(config-if-range)#switchport mode access
Switch2(config-if-range)#switchport access vlan 4
Switch2(config-if-range)#switchport voice vlan 4
Switch2(config-if-range)#exit
Switch2(config)#
Switch2(config)#interface fastEthernet 0/0//соединение с маршрутизатором
Switch2(config-if)#switchport mode trunk
Switch2(config-if)#exit
Switch2(config)#
Switch2(config)#exit
Switch2#wr

```

Коммутаторы имеют разделение по vlan'ам, теперь переходим к настройке маршрутизатора vIOS.

```

Router>en
Router#conf t
Router(config)#
Router(config)#interface fa0/0-1 //для обоих коммутаторов
Router(config-if)#no shutdown //поднимаем порты

```

```
Router(config-if)#exit
Router(config)#
Router(config)#interface fa0/0.10
Router(config-subif)#encapsulation dot1Q
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
Router(config-subif)#exit
Router(config)#
Router(config)#interface fa0/1.10
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
Router(config-subif)#exit
Router(config)#
Router(config)#interface fa0/0.20
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 192.168.20.1 255.255.255.0
Router(config-subif)#exit
Router(config)#
Router(config)#interface fa0/1.20
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 192.168.20.1 255.255.255.0 55
Router(config-subif)#exit
```

Приложение 2.